

FOI b7c b7d

FIG. 1

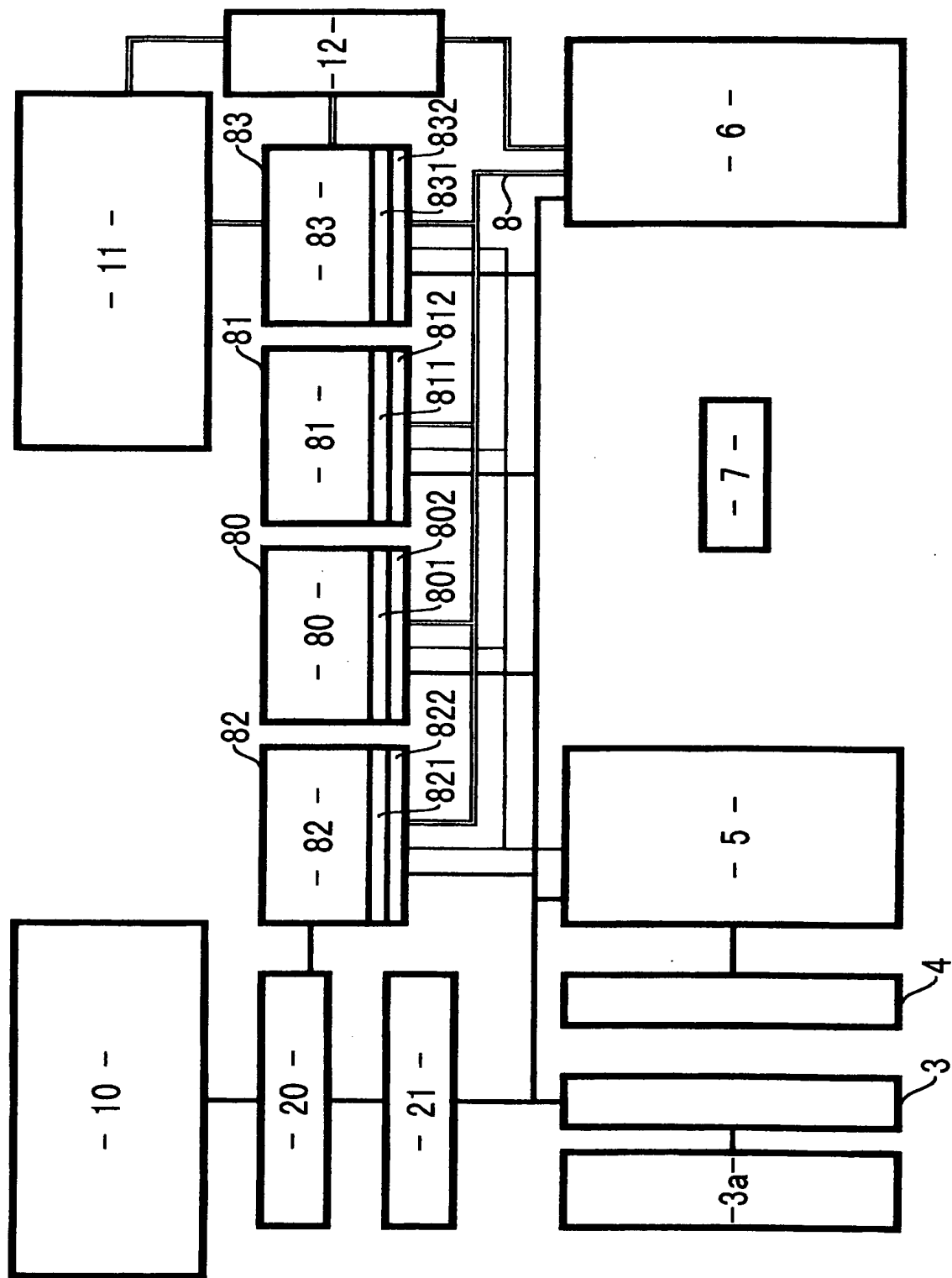


FIG.2

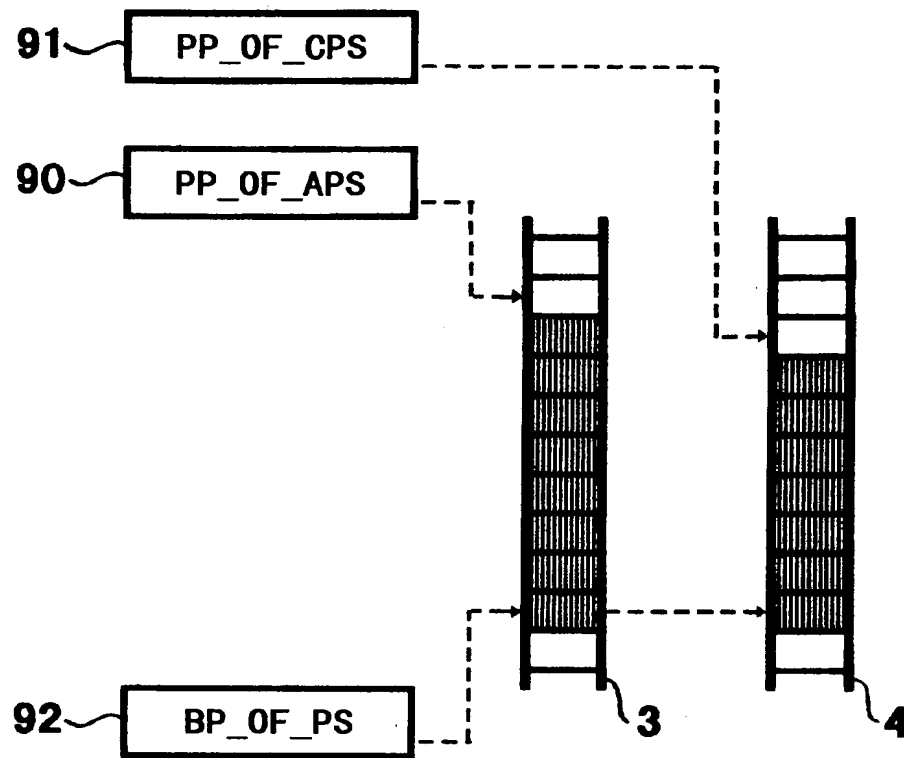


FIG.3

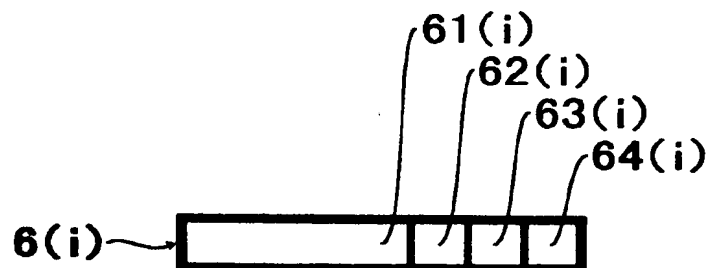


FIG. 4

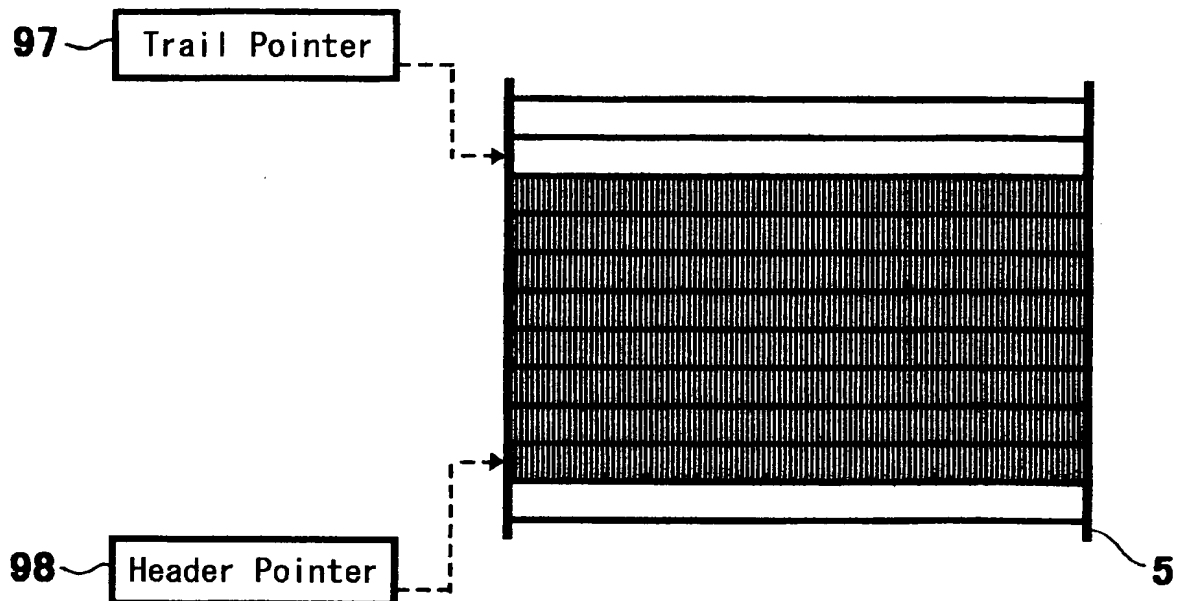


FIG. 5

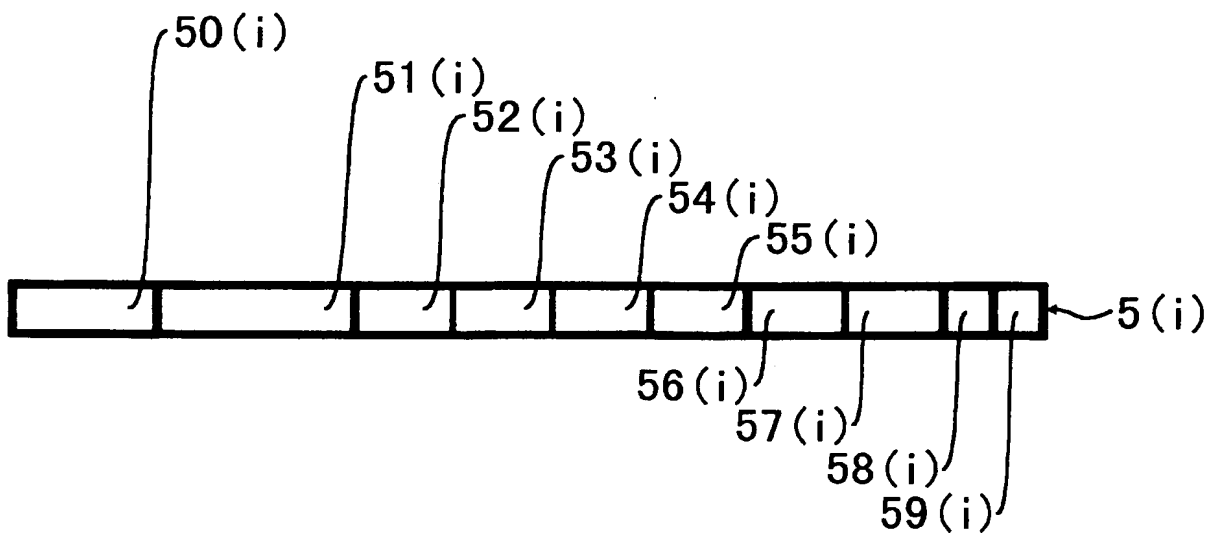


FIG.6

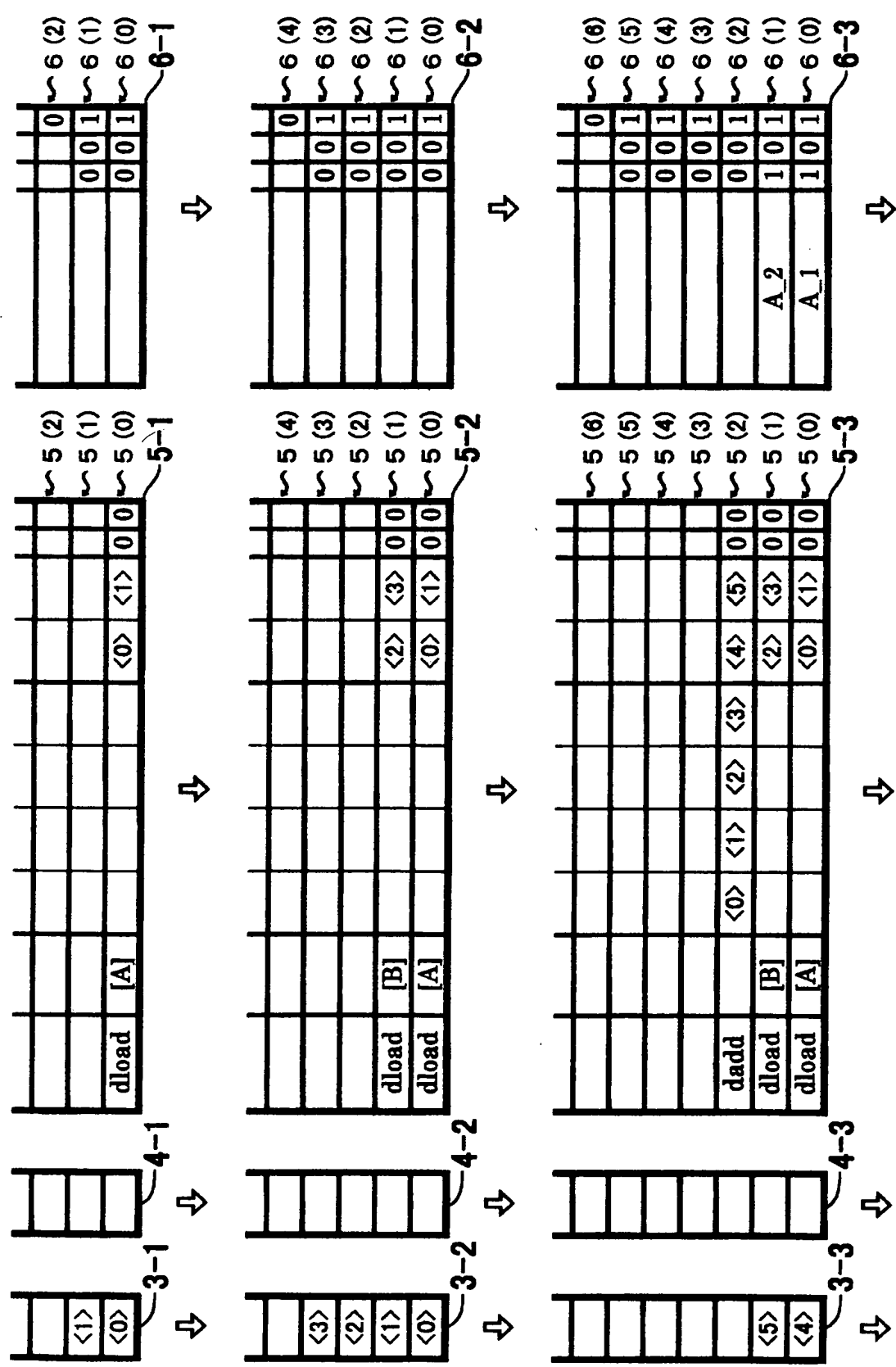


FIG. 7

FIG. 7

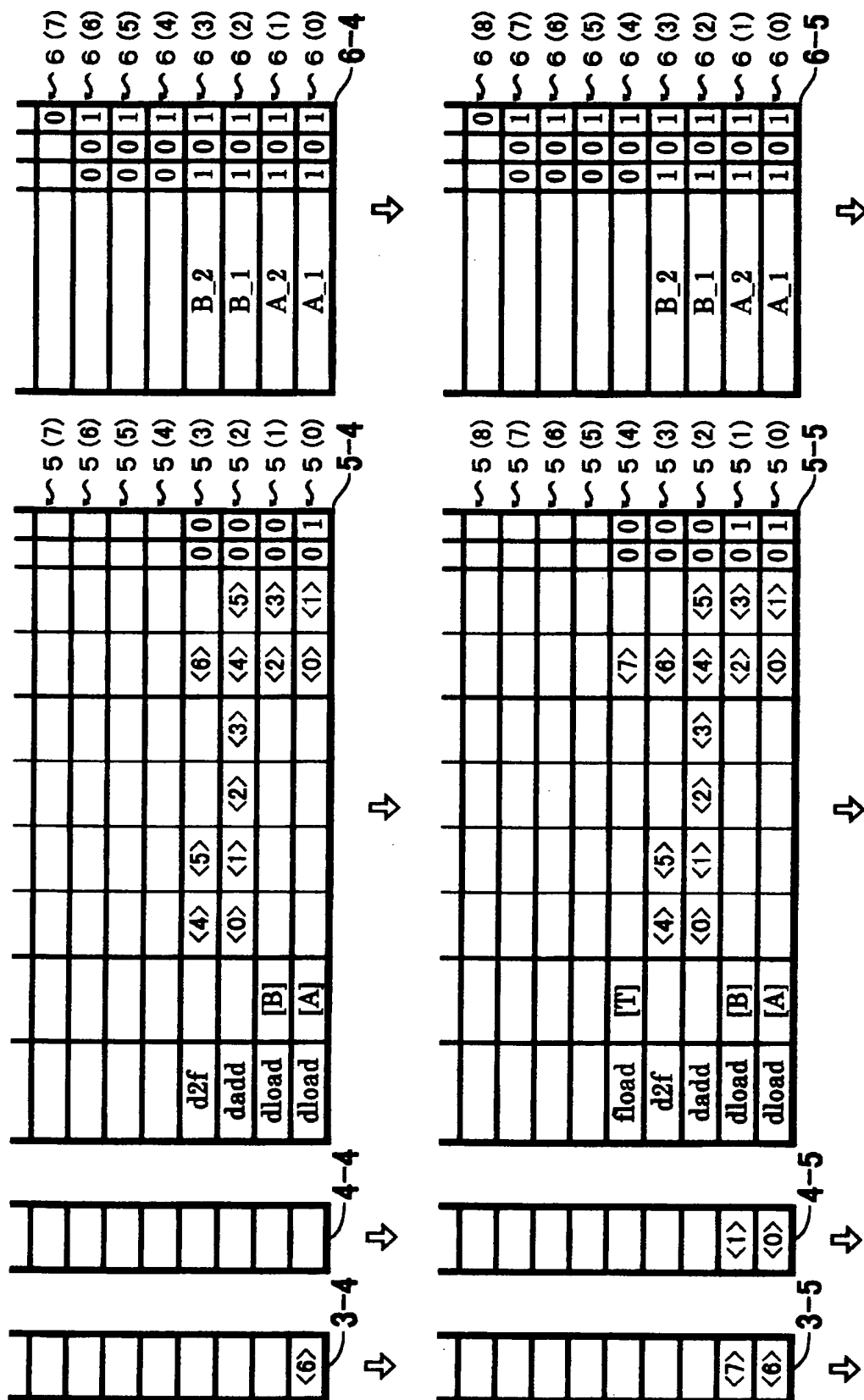


Figure 1 illustrates the execution of a program on a 6-processor system. The sequence of diagrams shows the state of the system at various points in time, with processors and memory blocks labeled 0-5 and 6-11 respectively. The sequence includes instructions like swap, fload, d2f, dadd, dload, and dload, and shows the flow of data between processors and memory blocks.

Figure 1 illustrates the state of a 1024-word memory array at various stages of execution. The array is divided into four sections: 3-8, 4-9, 5-8, and 6-9. Each section contains a grid of data. The top section (3-8) shows the initial state with values like 'fsub', 'dup_x1', 'swap', 'fload', 'd2f', 'dadd', 'dload', and 'dload'. The middle section (4-9) shows the state after the first stage of execution, with values like 'fdiv', 'fsub', 'dup_x1', 'swap', 'fload', 'd2f', 'dadd', 'dload', and 'dload'. The bottom section (5-8) shows the state after the second stage of execution, with values like 'fsub', 'dup_x1', 'swap', 'fload', 'd2f', 'dadd', 'dload', and 'dload'. The top-right section (6-9) shows the state after the third stage of execution, with values like 'fsub', 'dup_x1', 'swap', 'fload', 'd2f', 'dadd', 'dload', and 'dload'. Arrows indicate the flow of data between these stages.

FIG. 10

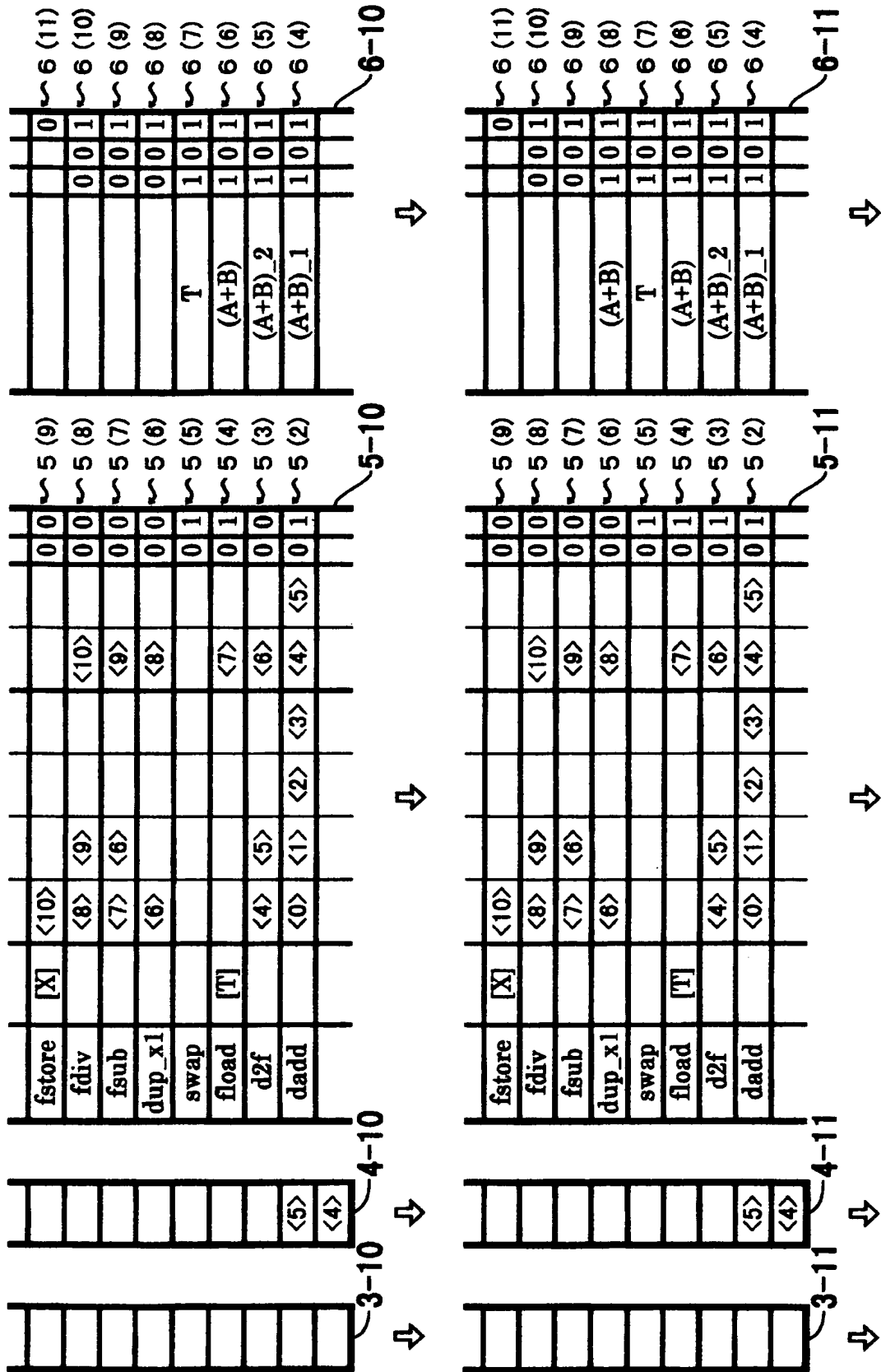


FIG. 11

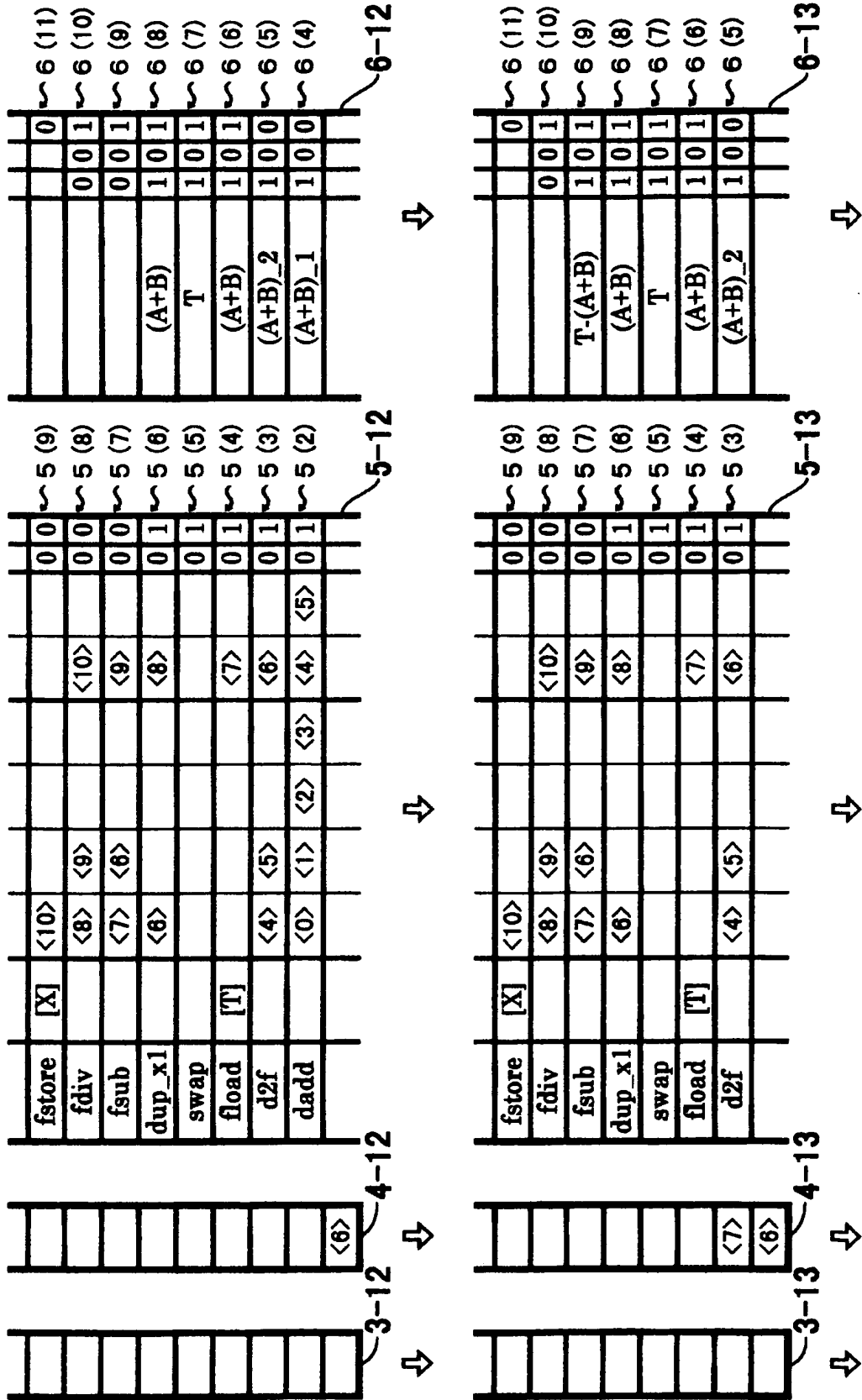


FIG. 12

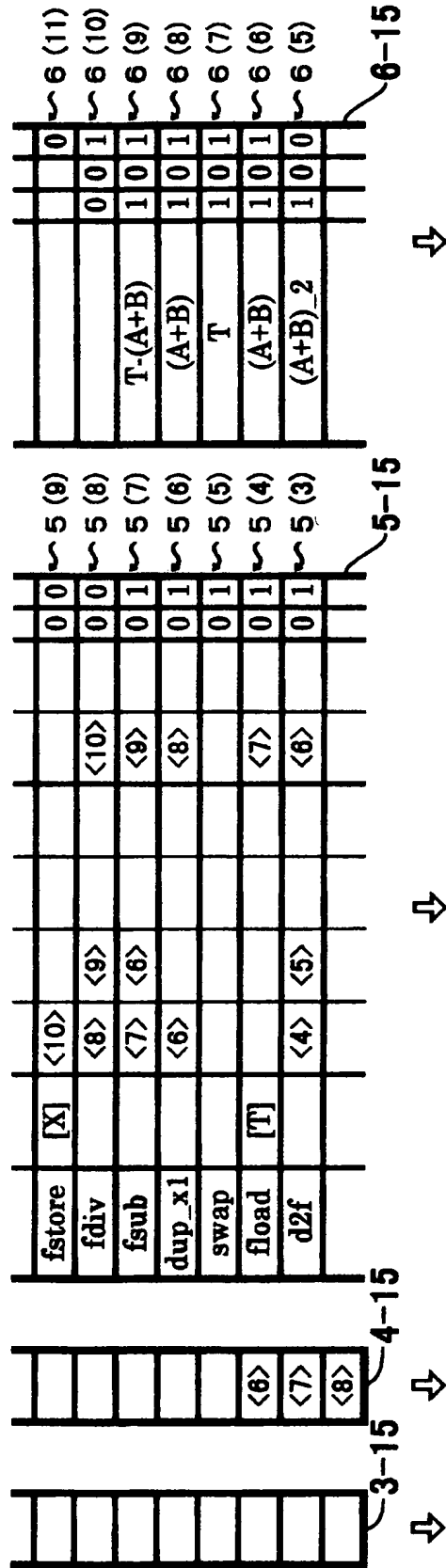
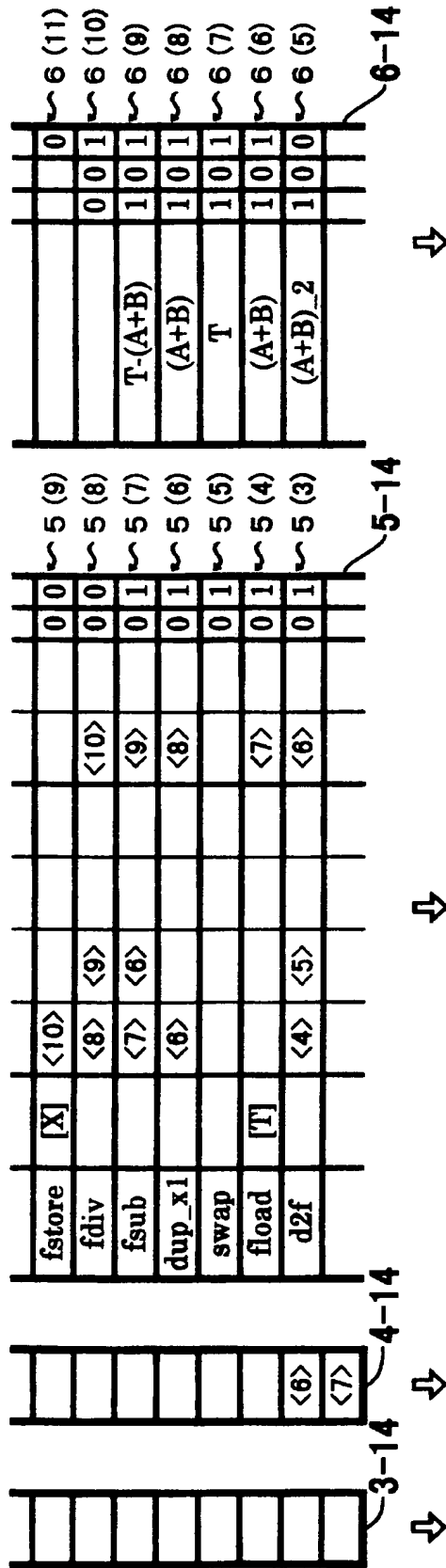


FIG. 13



FIG. 14

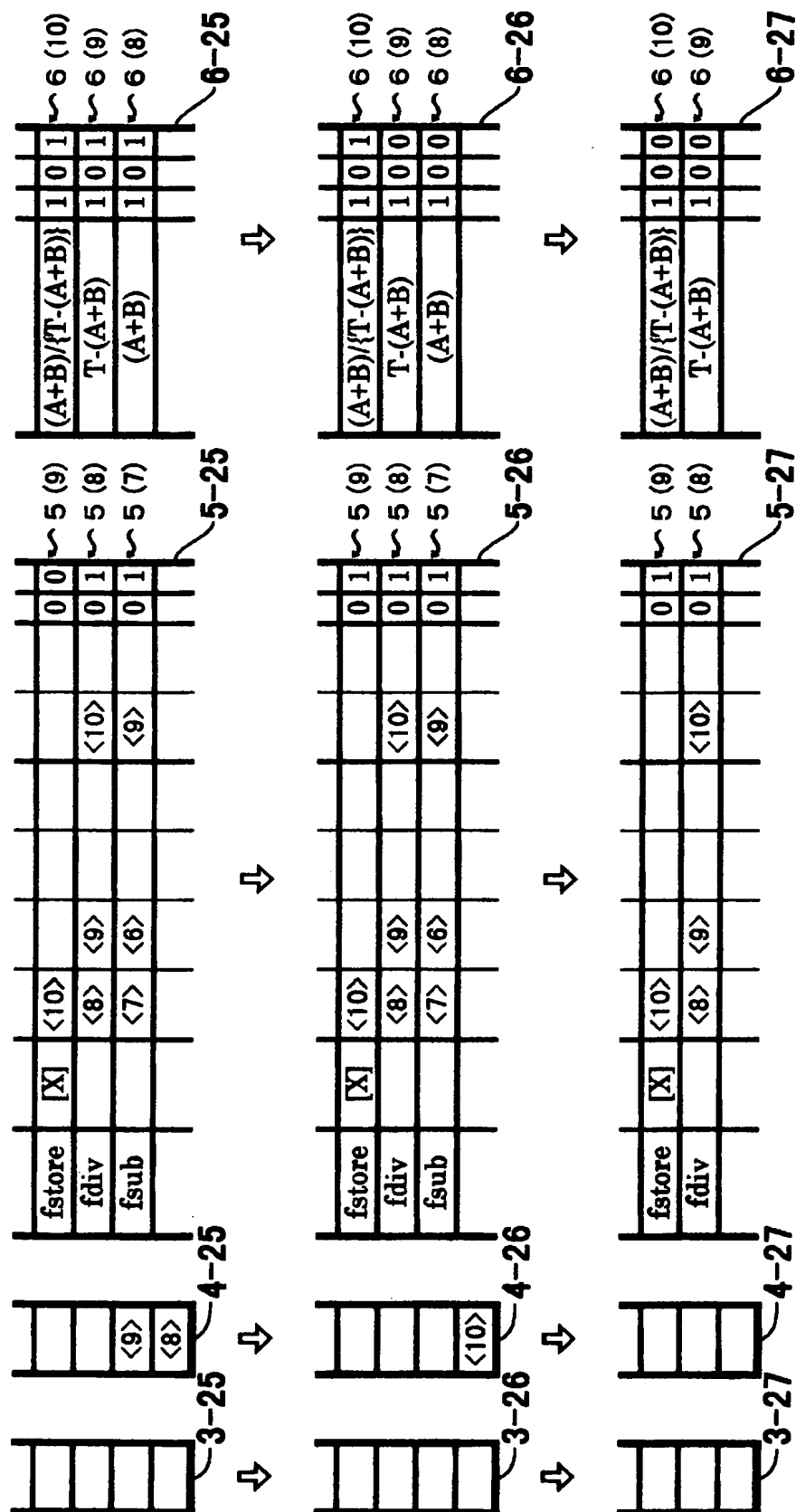


FIG.15

INCREASE IN PP_OF_APS	MANIPULATION TO BE APPLIED ON APS	SUBSTANCES TO BE WRITTEN INTO THREE IQ ENTRIES																						
+4	<table><tr><td></td><td>NC</td><td>f1</td><td>f2</td><td>f3</td><td>f4</td></tr></table>		NC	f1	f2	f3	f4	<table><tr><td>dload</td><td>[A]</td><td></td><td></td><td></td><td></td><td>f1</td><td>f2</td></tr><tr><td>dload</td><td>[B]</td><td></td><td></td><td></td><td></td><td>f3</td><td>f4</td></tr></table>	dload	[A]					f1	f2	dload	[B]					f3	f4
	NC	f1	f2	f3	f4																			
dload	[A]					f1	f2																	
dload	[B]					f3	f4																	
-3	<table><tr><td></td><td>NC</td><td>NC</td><td>NC</td><td>NC</td><td>f3</td></tr></table>		NC	NC	NC	NC	f3	<table><tr><td>dadd</td><td></td><td>s3</td><td>s2</td><td>s1</td><td>s0</td><td>f1</td><td>f2</td></tr><tr><td>d2f</td><td></td><td>f1</td><td>f2</td><td></td><td></td><td>f3</td><td></td></tr></table>	dadd		s3	s2	s1	s0	f1	f2	d2f		f1	f2			f3	
	NC	NC	NC	NC	f3																			
dadd		s3	s2	s1	s0	f1	f2																	
d2f		f1	f2			f3																		
+1	<table><tr><td></td><td>NC</td><td>NC</td><td>NC</td><td>f1</td><td>s0</td></tr></table>		NC	NC	NC	f1	s0	<table><tr><td>fload</td><td>[T]</td><td></td><td></td><td></td><td></td><td>f1</td><td></td></tr><tr><td>swap</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>	fload	[T]					f1		swap							
	NC	NC	NC	f1	s0																			
fload	[T]					f1																		
swap																								
0	<table><tr><td></td><td>NC</td><td>NC</td><td>NC</td><td>f1</td><td>f2</td></tr></table>		NC	NC	NC	f1	f2	<table><tr><td>dup_x1</td><td></td><td>s0</td><td></td><td></td><td></td><td>f1</td><td></td></tr><tr><td>fsub</td><td></td><td>s1</td><td>s0</td><td></td><td></td><td>f2</td><td></td></tr></table>	dup_x1		s0				f1		fsub		s1	s0			f2	
	NC	NC	NC	f1	f2																			
dup_x1		s0				f1																		
fsub		s1	s0			f2																		
-2	<table><tr><td></td><td>NC</td><td>NC</td><td>NC</td><td>NC</td><td>NC</td></tr></table>		NC	NC	NC	NC	NC	<table><tr><td>fdiv</td><td></td><td>s1</td><td>s0</td><td></td><td></td><td>f1</td><td></td></tr><tr><td>fstore</td><td>[X]</td><td>f1</td><td></td><td></td><td></td><td></td><td></td></tr></table>	fdiv		s1	s0			f1		fstore	[X]	f1					
	NC	NC	NC	NC	NC																			
fdiv		s1	s0			f1																		
fstore	[X]	f1																						

09/926320-101503